

# Notes on Kalman Filter (KF, EKF, ESKF, IEKF, IESKF)

Gyubeom Edward Im\*

February 8, 2024

## Contents

<b>1 Recursive bayes filter</b>	<b>2</b>
1.1 Derivation of recursive bayes filter . . . . .	2
1.2 Gaussian belief case . . . . .	4
<b>2 Kalman filter (KF)</b>	<b>4</b>
2.1 Prediction step . . . . .	5
2.2 Correction step . . . . .	5
2.3 Summary . . . . .	6
<b>3 Extended kalman filter (EKF)</b>	<b>6</b>
3.1 Prediction step . . . . .	7
3.2 Correction step . . . . .	7
3.3 Summary . . . . .	8
<b>4 Error-state kalman filter (ESKF)</b>	<b>8</b>
4.1 Prediction step . . . . .	10
4.2 Correction step . . . . .	11
4.2.1 Reset . . . . .	11
4.3 Summary . . . . .	11
<b>5 Iterated extended kalman filter (IEKF)</b>	<b>12</b>
5.1 Compare to EKF . . . . .	12
5.1.1 Commonality 1 . . . . .	12
5.1.2 Commonality 2 . . . . .	12
5.1.3 Commonality 3 . . . . .	12
5.1.4 Difference 1 . . . . .	12
5.1.5 Difference 2 . . . . .	13
5.2 Prediction step . . . . .	13
5.3 Correction step . . . . .	13
5.4 Summary . . . . .	13
<b>6 Iterated error-state kalman filter (IESKF)</b>	<b>14</b>
<b>7 Relation between Gauss-Newton and IEKF</b>	<b>14</b>
<b>8 Wrap-up</b>	<b>14</b>
8.1 Kalman Filter (KF) . . . . .	15
8.2 Extended Kalman Filter (EKF) . . . . .	15
8.3 Error-state Kalman Filter (ESKF) . . . . .	16
8.4 Iterated Extended Kalman Filter (IEKF) . . . . .	16
8.5 Iterated Error-state Kalman Filter (IESKF) . . . . .	17

---

\*blog: alida.tistory.com, email: gyurse@gmail.com

# 1 Recursive bayes filter

본 포스트는 필자가 Kalman filter 공부하면서 배운 내용을 정리한 포스트이다. Kalman filter를 이해하는데 필요한 기반 지식은 확률 이론(Probability Theory) 개념 정리 포스팅을 참조하면 된다. Particle filter에 대해 알고 싶으면 파티클 필터(Particle Filter) 개념 정리 포스팅을 참조하면 된다.

상태 추정 문제란 일반적으로 주어진 제어입력과 관측값이 있을 때 이를 바탕으로 현재 상태(e.g., 위치, 속도, 회전 등)를 추정하는 것을 의미한다. 제어입력과 관측값이 주어졌을 때 현재 상태  $\mathbf{x}_t$ 의 믿을만한 정도, 또는  $\mathbf{x}_t$ 에 대한 Belief  $bel(\mathbf{x}_t)$ 은 같이 정의한다.

$$bel(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \quad (1)$$

-  $\mathbf{x}_t$  : t 시간에서 상태 변수 -  $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$  : 1 t 시간에서 관측값 -  $\mathbf{u}_{1:t} = \{\mathbf{u}_1, \dots, \mathbf{u}_t\}$  : 1 t 시간에서 제어입력 -  $bel(\mathbf{x}_t)$  : Belief of  $\mathbf{x}_t$ 라고 불리며 시작 시간부터  $t$ 초까지 센서를 통한 관측  $\mathbf{z}_{1:t}$ 과 제어입력  $\mathbf{u}_{1:t}$ 으로 인해 현재 로봇이  $\mathbf{x}_t$ 에 위치할 확률(믿을만한 정도)을 의미

$bel(\cdot)$ 은 bayesian rule에 따라 표현되고 전개되기 때문에 bayes filter라고도 불린다. 위 식을 markov assumption과 bayesian rule을 사용하여 전개하면 재귀적인 필터를 유도할 수 있고 이를 recursive bayes filter라고 한다.

$$\begin{aligned} bel(\mathbf{x}_t) &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \overline{bel}(\mathbf{x}_t) \\ \overline{bel}(\mathbf{x}_t) &= \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \end{aligned} \quad (2)$$

-  $\eta = 1/p(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$  : 확률분포의 넓이를 1로 정규화하여 확률분포의 정의를 유지시켜주는 값 -  $p(\mathbf{z}_t \mid \mathbf{x}_t)$  : 관측 모델(observation model) -  $\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) d\mathbf{x}_{t-1}$  : 모션 모델(motion model)

Recursive bayes filter는 위와 같이 이전 스텝의  $bel(\mathbf{x}_{t-1})$ 로부터 현재 스텝의  $bel(\mathbf{x}_t)$ 를 구할 수 있으므로 재귀 필터라고 부른다.

## 1.1 Derivation of recursive bayes filter

Recursive bayes filter의 수식은 다음과 같이 유도된다.

$$\begin{aligned} bel(\mathbf{x}_t) &= p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1} \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot \overline{bel}(\mathbf{x}_{t-1}) \end{aligned} \quad (3)$$

**Step 1:**

$$\begin{aligned} bel(\mathbf{x}_t) &= p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \end{aligned} \quad (4)$$

Bayesian rule을 적용한다.  $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$

**Step 2:**

$$\begin{aligned} bel(\mathbf{x}_t) &= p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\ &= \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \end{aligned} \quad (5)$$

현재 상태는 바로 이전 상태에만 의존성을 지니는 Markov Assumption을 적용한다.

**Step 3:**

$$\begin{aligned}
 \text{bel}(\mathbf{x}_t) &= p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1}
 \end{aligned} \tag{6}$$

종합의 법칙(Law of total probability) 또는 Marginalization를 적용한다.  $p(x) = \int_y p(x|y) \cdot p(y) dy$

**Step 4:**

$$\begin{aligned}
 \text{bel}(\mathbf{x}_t) &= p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1}
 \end{aligned} \tag{7}$$

현재 상태는 바로 이전 상태에만 의존성을 지니는 Markov Assumption을 적용한다.

**Step 5:**

$$\begin{aligned}
 \text{bel}(\mathbf{x}_t) &= p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1}
 \end{aligned} \tag{8}$$

현재 상태는 바로 이전 상태에만 의존성을 지니는 Markov Assumption을 적용한다.

**Step 6:**

$$\begin{aligned}
 \text{bel}(\mathbf{x}_t) &= p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1} \\
 &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot \text{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}
 \end{aligned} \tag{9}$$

$p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})$ 은  $\text{bel}(\mathbf{x}_{t-1})$ 의 정의와 동일하므로 치환한다.

### Step 7:

$$\begin{aligned}
\text{bel}(\mathbf{x}_t) &= p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1} \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot \text{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \\
&= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \cdot \overline{\text{bel}}(\mathbf{x}_{t-1})
\end{aligned} \tag{10}$$

$\int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot \text{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$  을  $\overline{\text{bel}}(\mathbf{x}_{t-1})$ 로 치환한다.

## 1.2 Gaussian belief case

$\text{bel}(\mathbf{x}_t)$ 가 가우시안 분포를 따르는 경우 이를 특별히 칼만 필터(kalman filter)라고 한다.

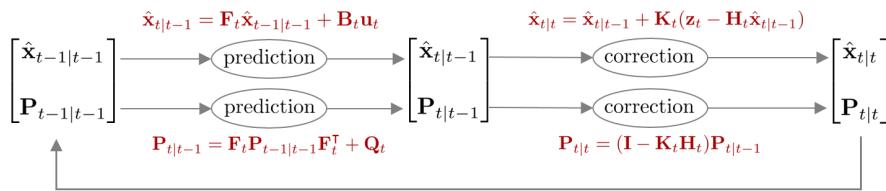
$$\begin{aligned}
\overline{\text{bel}}(\mathbf{x}_t) &\sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \quad (\text{Kalman Filter Prediction}) \\
\text{bel}(\mathbf{x}_t) &\sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \quad (\text{Kalman Filter Correction})
\end{aligned} \tag{11}$$

평균과 분산은  $(\hat{\mathbf{x}}, \mathbf{P})$  또는  $(\hat{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$ 로 표현하기도 한다. 이는 표기법만 다를 뿐 동일한 의미를 지닌다.

## 2 Kalman filter (KF)

### NOMENCLATURE of kalman filter

- 스칼라는 일반 소문자로 표기한다 e.g., a
- 벡터는 굵은 소문자로 표기한다 e.g., a
- 행렬은 굵은(bold) 대문자로 표기한다 e.g., R
- prediction:  $\overline{\text{bel}}(\mathbf{x}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1})$ 
  - $\hat{\mathbf{x}}_{t|t-1}$  :  $t-1$  스텝의 correction 값이 주어졌을 때  $t$  스텝의 평균. 일부 문헌은  $\mathbf{x}_t^-$ 로도 표기함.
  - $\hat{\mathbf{P}}_{t|t-1}$  :  $t-1$  스텝의 correction 값이 주어졌을 때  $t$  스텝의 공분산. 일부 문헌은  $\mathbf{P}_t^-$ 로도 표기함.
- correction:  $\text{bel}(\mathbf{x}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t})$ 
  - $\hat{\mathbf{x}}_{t|t}$  :  $t$  스텝의 prediction 값이 주어졌을 때  $t$  스텝의 평균. 일부 문헌은  $\mathbf{x}_t^+$ 로도 표기함.
  - $\hat{\mathbf{P}}_{t|t}$  :  $t$  스텝의 prediction 값이 주어졌을 때  $t$  스텝의 공분산. 일부 문헌은  $\mathbf{P}_t^+$ 로도 표기함.



시간  $t$ 에 로봇의 위치를  $\mathbf{x}_t$ , 로봇의 센서로 부터 관측한 값을  $\mathbf{z}_t$ , 로봇의 제어입력을  $\mathbf{u}_t$ 라고 하면 이를 통해 모션 모델(motion model)과 관측 모델(observation model)을 정의할 수 있다. 이 때, 모션 모델과 관측 모델은 선형이어야(linear model) 한다는 제약조건이 있다. 모션 모델과 관측 모델은 다음과 같다.

$$\begin{aligned} \text{Motion Model: } \mathbf{x}_t &= \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t \\ \text{Observation Model: } \mathbf{z}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t \end{aligned} \quad (12)$$

- $\mathbf{x}_t$ : 모델의 상태 변수(state variable)
- $\mathbf{u}_t$ : 모델의 입력(input)
- $\mathbf{z}_t$ : 모델의 관측값(measurement)
- $\mathbf{F}_t$ : 모델의 상태전이(state transition) 행렬
- $\mathbf{B}_t$ : 모델에 입력  $\mathbf{u}_t$ 가 주어졌을 때  $\mathbf{u}_t$ 를 상태 변수로 변환해주는 행렬
- $\mathbf{H}_t$ : 모델의 관측(observation) 행렬
- $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ : 모션 모델의 노이즈.  $\mathbf{Q}_t$ 는  $\mathbf{w}_t$ 의 공분산 행렬을 의미한다.
- $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ : 관측 모델의 노이즈.  $\mathbf{R}_t$ 는  $\mathbf{v}_t$ 의 공분산 행렬을 의미한다.

확률변수가 모두 가우시안 분포를 따른다고 가정하면  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t), p(\mathbf{z}_t | \mathbf{x}_t)$ 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) &\sim \mathcal{N}(\mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t, \mathbf{Q}_t) \\ &= \frac{1}{\sqrt{\det(2\pi \mathbf{Q}_t)}} \exp \left( -\frac{1}{2} (\mathbf{x}_t - \mathbf{F}_t \mathbf{x}_{t-1} - \mathbf{B}_t \mathbf{u}_t)^T \mathbf{Q}_t^{-1} (\mathbf{x}_t - \mathbf{F}_t \mathbf{x}_{t-1} - \mathbf{B}_t \mathbf{u}_t) \right) \end{aligned} \quad (13)$$

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{x}_t) &\sim \mathcal{N}(\mathbf{H}_t \mathbf{x}_t, \mathbf{R}_t) \\ &= \frac{1}{\sqrt{\det(2\pi \mathbf{R}_t)}} \exp \left( -\frac{1}{2} (\mathbf{z}_t - \mathbf{H}_t \mathbf{x}_t)^T \mathbf{R}_t^{-1} (\mathbf{z}_t - \mathbf{H}_t \mathbf{x}_t) \right) \end{aligned} \quad (14)$$

다음으로 칼만 필터를 통해 구해야 하는  $\overline{\text{bel}}(\mathbf{x}_t), \text{bel}(\mathbf{x}_t)$ 은 아래와 같이 나타낼 수 있다.

$$\begin{aligned} \overline{\text{bel}}(\mathbf{x}_t) &= \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \text{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \\ \text{bel}(\mathbf{x}_t) &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \overline{\text{bel}}(\mathbf{x}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \end{aligned} \quad (15)$$

칼만 필터는 prediction에서 이전 스텝의 값과 모션 모델을 사용하여 예측값  $\overline{\text{bel}}(\mathbf{x}_t)$ 을 먼저 구한 후 correction에서 관측값과 관측 모델을 사용하여 보정된 값  $\text{bel}(\mathbf{x}_t)$ 을 구하는 방식으로 동작한다. 초기값  $\text{bel}(\mathbf{x}_0)$ 은 다음과 같이 주어진다.

$$\text{bel}(\mathbf{x}_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \quad (16)$$

- $\hat{\mathbf{x}}_0$ : 일반적으로 0으로 설정한다
- $\mathbf{P}_0$ : 일반적으로 작은 값( $< 1e-2$ )으로 설정한다.

## 2.1 Prediction step

Prediction은  $\overline{\text{bel}}(\mathbf{x}_t)$ 를 구하는 과정을 말한다.  $\overline{\text{bel}}(\mathbf{x}_t)$ 는 가우시안 분포를 따르므로 평균  $\hat{\mathbf{x}}_{t|t-1}$ 과 분산  $\mathbf{P}_{t|t-1}$ 을 각각 구해보면 아래와 같이 구할 수 있다.

$$\boxed{\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_t \end{aligned}} \quad (17)$$

## 2.2 Correction step

Correction은  $\text{bel}(\mathbf{x}_t)$ 를 구하는 과정을 말한다.  $\text{bel}(\mathbf{x}_t)$  또한 가우시안 분포를 따르므로 평균  $\hat{\mathbf{x}}_{t|t}$ 과 분산  $\mathbf{P}_{t|t}$ 을 각각 구해보면 다음과 같다. 이 때,  $\mathbf{K}_t$ 는 칼만 게인(kalman gain)을 의미한다.

$$\boxed{\begin{aligned} \mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t)^{-1} \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}) \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \end{aligned}} \quad (18)$$

## 2.3 Summary

칼만 필터를 함수로 표현하면 다음과 같다.

```

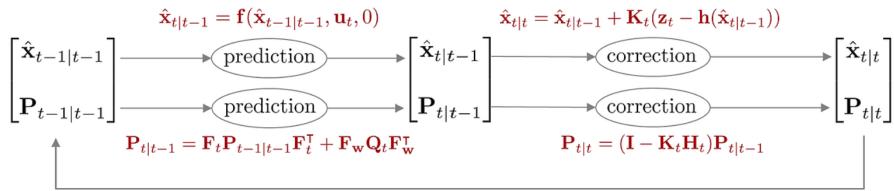
KalmanFilter( $\hat{\mathbf{x}}_{t-1|t-1}$ ,  $\mathbf{P}_{t-1|t-1}$ ,  $\mathbf{u}_t$ ,  $\mathbf{z}_t$ ) {
    (Prediction Step)
     $\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t$ 
     $\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t$ 

    (Correction Step)
     $\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_t)^{-1}$ 
     $\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1})$ 
     $\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}$ 
    return  $\hat{\mathbf{x}}_{t|t}$ ,  $\mathbf{P}_{t|t}$ 
}

```

(19)

## 3 Extended kalman filter (EKF)



칼만 필터는 모션 모델과 관측 모델이 선형이라는 가정 하에 상태를 추정한다. 하지만 현실세계의 대부분의 현상들은 비선형으로 모델링되므로 앞서 정의한 칼만 필터를 그대로 적용하면 정상적으로 동작하지 않는다. 비선형의 모션 모델과 관측 모델에서도 칼만 필터를 사용하기 위해 확장칼만필터(exended kalman filter, EKF)가 제안되었다. **EKF는 테일러 1차 근사(talyor 1st approximation)을 사용하여 비선형 모델을 선형 모델로 근사한 후 칼만 필터를 적용하는 방법을 사용한다.** EKF의 모션 모델과 관측 모델은 다음과 같다.

$$\begin{aligned}
\text{Motion Model: } & \mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{w}_t) \\
\text{Observation Model: } & \mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t)
\end{aligned} \tag{20}$$

- $\mathbf{x}_t$ : 모델의 상태 변수(state variable)
- $\mathbf{u}_t$ : 모델의 입력(input)
- $\mathbf{z}_t$ : 모델의 관측값(measurement)
- $\mathbf{f}(\cdot)$ : 비선형 모션(motion) 모델 함수
- $\mathbf{h}(\cdot)$ : 비선형 관측(observation) 모델 함수
- $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ : 모션 모델의 노이즈.  $\mathbf{Q}_t$ 는  $\mathbf{w}_t$ 의 공분산 행렬을 의미
- $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ : 관측 모델의 노이즈.  $\mathbf{R}_t$ 는  $\mathbf{v}_t$ 의 공분산 행렬을 의미

위 식에서  $\mathbf{f}(\cdot)$ 은 비선형 모션 모델을 의미하고  $\mathbf{h}(\cdot)$ 은 비선형 관측 모델을 의미한다.  $\mathbf{f}(\cdot), \mathbf{h}(\cdot)$ 에 각각 1차 테일러 근사를 수행하면 다음과 같다.

$$\begin{aligned}
\mathbf{x}_t &\approx \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) + \mathbf{F}_t (\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1|t-1}) + \mathbf{F}_w \mathbf{w}_t \\
\mathbf{z}_t &\approx \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) + \mathbf{H}_t (\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) + \mathbf{H}_v \mathbf{v}_t
\end{aligned} \tag{21}$$

이 때,  $\mathbf{F}_t$ 는  $\hat{\mathbf{x}}_{t-1|t-1}$ 에서 계산한 모션 모델의 자코비안 행렬을 의미하며  $\mathbf{H}_t$ 는  $\hat{\mathbf{x}}_{t|t-1}$ 에서 계산한 관측 모델의 자코비안 행렬을 의미한다. 그리고  $\mathbf{F}_w, \mathbf{H}_v$ 는 각각  $\mathbf{w}_t = 0, \mathbf{v}_t = 0$ 에서 노이즈에 대한 자코비안 행렬을 의미한다. 자코비안에 대한 자세한 내용은 해당 포스트를 참조하면된다.

$$\mathbf{F}_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_{t-1}} \Big|_{\mathbf{x}_{t-1}=\hat{\mathbf{x}}_{t-1|t-1}} \quad \mathbf{F}_w = \frac{\partial \mathbf{f}}{\partial \mathbf{w}_t} \Big|_{\substack{\mathbf{x}_{t-1}=\hat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{w}_t=0}} \tag{22}$$

$$\mathbf{H}_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_t} \Big|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}} \quad \mathbf{H}_{\mathbf{v}} = \frac{\partial \mathbf{h}}{\partial \mathbf{v}_t} \Big|_{\substack{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1} \\ \mathbf{v}_t=0}} \quad (23)$$

(21) 식을 전개하면 다음과 같다.

$$\boxed{\begin{aligned} \mathbf{x}_t &= \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) - \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{F}_{\mathbf{w}} \mathbf{w}_t \\ &= \mathbf{F}_t \mathbf{x}_{t-1} + \tilde{\mathbf{u}}_t + \tilde{\mathbf{w}}_t \end{aligned}} \quad (24)$$

-  $\tilde{\mathbf{u}}_t = \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) - \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1}$  -  $\tilde{\mathbf{w}}_t = \mathbf{F}_{\mathbf{w}} \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{F}_{\mathbf{w}} \mathbf{Q}_t \mathbf{F}_{\mathbf{w}}^T)$

$$\boxed{\begin{aligned} \mathbf{z}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} + \mathbf{H}_{\mathbf{v}} \mathbf{v}_t \\ &= \mathbf{H}_t \mathbf{x}_t + \tilde{\mathbf{z}}_t + \tilde{\mathbf{v}}_t \end{aligned}} \quad (25)$$

-  $\tilde{\mathbf{z}}_t = \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}$  -  $\tilde{\mathbf{v}}_t = \mathbf{H}_{\mathbf{v}} \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{H}_{\mathbf{v}} \mathbf{R}_t \mathbf{H}_{\mathbf{v}}^T)$

확률변수가 모두 가우시안 분포를 따른다고 가정하면  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t), p(\mathbf{z}_t | \mathbf{x}_t)$ 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) &\sim \mathcal{N}(\mathbf{F}_t \mathbf{x}_{t-1} + \tilde{\mathbf{u}}_t, \mathbf{F}_{\mathbf{w}} \mathbf{Q}_t \mathbf{F}_{\mathbf{w}}^T) \\ &= \frac{1}{\sqrt{\det(2\pi \mathbf{F}_{\mathbf{w}} \mathbf{Q}_t \mathbf{F}_{\mathbf{w}}^T)}} \exp \left( -\frac{1}{2} (\mathbf{x}_t - \mathbf{F}_t \mathbf{x}_{t-1} - \tilde{\mathbf{u}}_t)^T (\mathbf{F}_{\mathbf{w}} \mathbf{Q}_t \mathbf{F}_{\mathbf{w}}^T)^{-1} (\mathbf{x}_t - \mathbf{F}_t \mathbf{x}_{t-1} - \tilde{\mathbf{u}}_t) \right) \end{aligned} \quad (26)$$

$$\begin{aligned} p(\mathbf{z}_t | \mathbf{x}_t) &\sim \mathcal{N}(\mathbf{H}_t \mathbf{x}_t + \tilde{\mathbf{z}}_t, \mathbf{H}_{\mathbf{v}} \mathbf{R}_t \mathbf{H}_{\mathbf{v}}^T) \\ &= \frac{1}{\sqrt{\det(2\pi \mathbf{H}_{\mathbf{v}} \mathbf{R}_t \mathbf{H}_{\mathbf{v}}^T)}} \exp \left( -\frac{1}{2} (\mathbf{z}_t - \mathbf{H}_t \mathbf{x}_t - \tilde{\mathbf{z}}_t)^T (\mathbf{H}_{\mathbf{v}} \mathbf{R}_t \mathbf{H}_{\mathbf{v}}^T)^{-1} (\mathbf{z}_t - \mathbf{H}_t \mathbf{x}_t - \tilde{\mathbf{z}}_t) \right) \end{aligned} \quad (27)$$

$\mathbf{F}_{\mathbf{w}} \mathbf{Q}_t \mathbf{F}_{\mathbf{w}}^T$ 은 선형화된 모션 모델의 노이즈를 의미하며  $\mathbf{H}_{\mathbf{v}} \mathbf{R}_t \mathbf{H}_{\mathbf{v}}^T$ 은 선형화된 관측 모델의 노이즈를 의미한다. 다음으로 칼만 필터를 통해 구해야 하는  $\overline{\text{bel}}(\mathbf{x}_t)$ ,  $\text{bel}(\mathbf{x}_t)$ 은 아래와 같이 나타낼 수 있다.

$$\begin{aligned} \overline{\text{bel}}(\mathbf{x}_t) &= \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \text{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \\ \text{bel}(\mathbf{x}_t) &= \eta \cdot p(\mathbf{z}_t | \mathbf{x}_t) \overline{\text{bel}}(\mathbf{x}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \end{aligned} \quad (28)$$

EKF 또한 KF와 동일하게 prediction에서 이전 스텝의 값과 모션 모델을 사용하여 예측값  $\overline{\text{bel}}(\mathbf{x}_t)$ 을 먼저 구한 후 correction에서 관측값과 관측 모델을 사용하여 보정된 값  $\text{bel}(\mathbf{x}_t)$ 을 구하는 방식으로 동작한다. 초기값  $\text{bel}(\mathbf{x}_0)$ 은 다음과 같이 주어진다.

$$\text{bel}(\mathbf{x}_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \quad (29)$$

- $\hat{\mathbf{x}}_0$ : 일반적으로 0으로 설정한다
- $\mathbf{P}_0$  : 일반적으로 작은 값( $< 1e-2$ )으로 설정한다.

### 3.1 Prediction step

Prediction은  $\overline{\text{bel}}(\mathbf{x}_t)$ 를 구하는 과정을 말한다. 공분산 행렬을 구할 때 선형화된 자코비안 행렬  $\mathbf{F}_t$ 가 사용된다.

$$\boxed{\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{F}_{\mathbf{w}} \mathbf{Q}_t \mathbf{F}_{\mathbf{w}}^T \end{aligned}} \quad (30)$$

### 3.2 Correction step

Correction은  $\text{bel}(\mathbf{x}_t)$ 를 구하는 과정을 말한다. 칼만 게인과 공분산 행렬을 구할 때 선형화된 자코비안 행렬  $\mathbf{H}_t$ 가 사용된다.

$$\boxed{\begin{aligned} \mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{H}_{\mathbf{v}} \mathbf{R}_t \mathbf{H}_{\mathbf{v}}^T)^{-1} \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0)) \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \end{aligned}} \quad (31)$$

### 3.3 Summary

확장칼만필터를 함수로 표현하면 다음과 같다.

```

ExtendedKalmanFilter( $\hat{\mathbf{x}}_{t-1|t-1}$ ,  $\mathbf{P}_{t-1|t-1}$ ,  $\mathbf{u}_t$ ,  $\mathbf{z}_t$ ) {
    (Prediction Step)
     $\hat{\mathbf{x}}_{t|t-1} = \mathbf{f}_t(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0)$ 
     $\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top$ 

    (Correction Step)
     $\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t + \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top)^{-1}$ 
     $\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{h}_t(\hat{\mathbf{x}}_{t|t-1}, 0))$ 
     $\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}$ 
    return  $\hat{\mathbf{x}}_{t|t}$ ,  $\mathbf{P}_{t|t}$ 
}

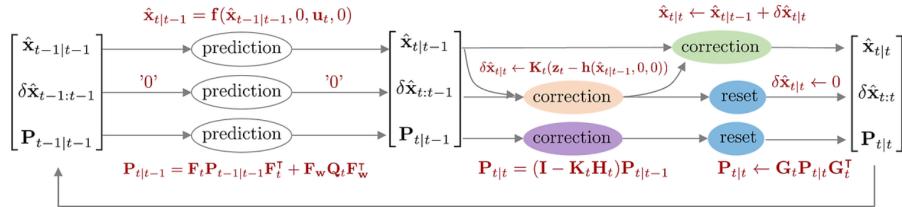
```

(32)

## 4 Error-state kalman filter (ESKF)

### NOMENCLATURE of error-state kalman filter

- prediction:  $\text{bel}(\delta\mathbf{x}_t) \sim \mathcal{N}(\delta\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1})$ 
  - $\delta\hat{\mathbf{x}}_{t|t-1}$ :  $t-1$  스텝의 correction 값이 주어졌을 때  $t$  스텝의 평균. 일부 문헌은  $\delta\mathbf{x}_t^-$ 로도 표기함.
  - $\hat{\mathbf{P}}_{t|t-1}$ :  $t-1$  스텝의 correction 값이 주어졌을 때  $t$  스텝의 공분산. 일부 문헌은  $\mathbf{P}_t^-$ 로도 표기함.
- correction:  $\text{bel}(\delta\mathbf{x}_t) \sim \mathcal{N}(\delta\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t})$ 
  - $\delta\hat{\mathbf{x}}_{t|t}$ :  $t$  스텝의 prediction 값이 주어졌을 때  $t$  스텝의 평균. 일부 문헌은  $\delta\mathbf{x}_t^+$ 로도 표기함.
  - $\hat{\mathbf{P}}_{t|t}$ :  $t$  스텝의 prediction 값이 주어졌을 때  $t$  스텝의 공분산. 일부 문헌은  $\mathbf{P}_t^+$ 로도 표기함.



에러상태 칼만필터(error-state kalman filter, ESKF)는 기존의 상태 변수  $\mathbf{x}_t$ 의 평균과 분산을 추정하는 EKF와 달리 에러 상태 변수  $\delta\mathbf{x}_t$ 의 평균과 분산을 추정하는 칼만 필터 알고리즘을 말한다. 상태 변수를 바로 추정하지 않고 에러상태를 통해 추정하기 때문에 indirect kalman filter라고도 불린다. 또 다른 이름으로는 error-state extended kalman filter(ES-EKF)로도 불린다. ESKF는 기존 상태변수를 true 상태 변수라고 부르며 이를 다음과 같이 명목(nominal) 상태와 에러(error) 상태의 합으로 표현한다.

$$\mathbf{x}_{\text{true},t} = \mathbf{x}_t + \delta\mathbf{x}_t \quad (33)$$

- $\mathbf{x}_{\text{true},t}$ : 기존 KF, EKF에서 업데이트된  $t$  스텝의 true 상태변수
- $\mathbf{x}_t$ :  $t$  스텝의 명목(nominal) 상태 변수. 에러가 없는 상태를 의미한다
- $\delta\mathbf{x}_t$ :  $t$  스텝의 에러(error) 상태 변수

위 식을 해석하면 실제(true) 추정하고자 하는 상태 변수  $\mathbf{x}_t$ 는 에러가 없는 일반(또는 명목, nominal) 상태  $\hat{\mathbf{x}}_t$ 과 모델 및 센서 노이즈로부터 발생하는 에러 상태  $\delta\mathbf{x}_t$ 의 합으로 나타낼 수 있다는 의미이다. 이 때, nominal 상태는 (상대적으로) 큰 값을 가지며 비선형성을 가진다. 반면에 에러 상태는 0 근처의 작은 값을 가지고 선형성을 가진다. **기존의 EKF는 비선형성이 큰 true (nominal + error) 상태 변수를 선형화하여 필터링하기 때문에 속도가 느리고 시간이 지날수록 에러 값이 누적되는 반면에, ESKF는 에러 상태만을 선형화하여 필터링하기 때문에 속도 및 정확성이 더욱 빠른 장점이 있다.** 기존 EKF와 비교했을 때 ESKF 이 가지는 장점들을 정리하면 다음과 같다(Madyastha et al., 2011):

- 방향(orientation)에 대한 여러 상태 표현법이 최소한의 파라미터를 가진다. 즉, 자유도만큼의 최소 파라미터를 가지기 때문에 over-parameterized로 인해 발생하는 특이점(signularity) 같은 현상이 발생하지 않는다.
- 여러 상태 시스템은 항상 원점(origin) 근처에서만 동작하기 때문에 선형화하기 용이하다. 따라서 점 별락 같은 파라미터 특이점 현상이 발생하지 않으며 항상 선형화를 수행할 수 있다.
- 여러 상태는 일반적으로 값이 작기 때문에 2차항 이상의 값들은 무시할 수 있다. 이는 자코비안 연산을 쉽고 빠르게 수행할 수 있도록 도와준다. 몇몇 자코비안은 상수화하여 사용하기도 한다.

다만 ESKF는 prediction 속도는 빠르지만 nominal 상태에서 일반적으로 비선형을 가진 큰 값들이 처리되므로 nominal 상태가 처리되는 correction 스텝의 속도가 느린 편이다.

ESKF의 모션 모델과 관측 모델은 다음과 같다.

$$\begin{aligned} \text{Error-state Motion Model:} \quad \mathbf{x}_t + \delta\mathbf{x}_t &= \mathbf{f}(\mathbf{x}_{t-1}, \delta\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{w}_t) \\ \text{Error-state Observation Model:} \quad \mathbf{z}_t &= \mathbf{h}(\mathbf{x}_t, \delta\mathbf{x}_t, \mathbf{v}_t) \end{aligned} \quad (34)$$

- $\mathbf{x}_t$ : 모델의 nominal 상태 변수
- $\delta\mathbf{x}_t$ : 모델의 에러 상태 변수
- $\mathbf{u}_t$ : 모델의 입력(input)
- $\mathbf{z}_t$ : 모델의 관측값(measurement)
- $\mathbf{f}(\cdot)$ : 비선형 모션(motion) 모델 함수
- $\mathbf{h}(\cdot)$ : 비선형 관측(observation) 모델 함수
- $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ : 에러 상태 모델의 노이즈.  $\mathbf{Q}_t$ 는  $\mathbf{w}_t$ 의 공분산 행렬을 의미
- $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ : 관측 모델의 노이즈.  $\mathbf{R}_t$ 는  $\mathbf{v}_t$ 의 공분산 행렬을 의미

$\mathbf{f}(\cdot), \mathbf{h}(\cdot)$ 에 각각 1차 테일러 근사를 수행하면 다음과 같다. 해당 전개는 [5]를 참고하여 작성하였다.

$$\begin{aligned} \mathbf{x}_t + \delta\mathbf{x}_t &\approx \mathbf{f}(\mathbf{x}_{t-1|t-1}, \delta\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) + \mathbf{F}_t(\delta\mathbf{x}_{t-1} - \delta\hat{\mathbf{x}}_{t-1|t-1}) + \mathbf{F}_w\mathbf{w}_t \\ \mathbf{z}_t &\approx \mathbf{h}(\mathbf{x}_{t|t-1}, \delta\hat{\mathbf{x}}_{t|t-1}, 0) + \mathbf{H}_t(\delta\mathbf{x}_t - \delta\hat{\mathbf{x}}_{t|t-1}) + \mathbf{H}_v\mathbf{v}_t \end{aligned} \quad (35)$$

이 때, 두 자코비안  $\mathbf{F}_t, \mathbf{H}_t$  모두 true 상태  $\mathbf{x}_{true,t}$ 가 아닌 에러 상태  $\delta\mathbf{x}_t$ 에 대한 자코비안임에 유의한다. 해당 자코비안 부분이 EKF와 가장 다른 부분이다

$$\boxed{\mathbf{F}_t = \frac{\partial \mathbf{f}}{\partial \delta\mathbf{x}_{t-1}} \Big|_{\delta\mathbf{x}_{t-1}=\delta\hat{\mathbf{x}}_{t-1|t-1}} \quad \mathbf{F}_w = \frac{\partial \mathbf{f}}{\partial \mathbf{w}_t} \Big|_{\delta\mathbf{x}_{t-1}=\delta\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{w}_t=0}} \quad (36)$$

$$\boxed{\mathbf{H}_t = \frac{\partial \mathbf{h}}{\partial \delta\mathbf{x}_t} \Big|_{\delta\mathbf{x}_t=\delta\hat{\mathbf{x}}_{t|t-1}} \quad \mathbf{H}_v = \frac{\partial \mathbf{h}}{\partial \mathbf{v}_t} \Big|_{\delta\mathbf{x}_t=\delta\hat{\mathbf{x}}_{t|t-1}, \mathbf{v}_t=0}} \quad (37)$$

$\mathbf{H}_t$ 는 다음과 같이 연쇄법칙을 통해 표현할 수 있다.

$$\mathbf{H}_t = \frac{\partial \mathbf{h}}{\partial \delta\mathbf{x}_t} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_{true,t}} \frac{\partial \mathbf{x}_{true,t}}{\partial \delta\mathbf{x}_t} \quad (38)$$

이 중 앞 부분  $\frac{\partial \mathbf{h}}{\partial \mathbf{x}_{true,t}}$ 은 EKF에서 구한 자코비안과 동일하지만 에러 상태 변수에 대한 자코비안  $\frac{\partial \mathbf{x}_{true,t}}{\partial \delta\mathbf{x}_t}$ 이 추가되었다. 이에 대한 자세한 내용은 Quaternion kinematics for the error-state Kalman filter 내용 정리 포스트를 참고하면 된다.

(35) 식을 전개하면 다음과 같다.

$$\mathbf{x}_t + \delta\mathbf{x}_t = \mathbf{F}_t \delta\mathbf{x}_{t-1} + \mathbf{f}(\mathbf{x}_{t-1|t-1}, \delta\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) - \mathbf{F}_t \delta\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{F}_w \mathbf{w}_t \quad (39)$$

이전 correction 스텝에서 항상  $\delta\hat{\mathbf{x}}_{t-1|t-1} = 0$ 으로 초기화되므로 관련된 항에 0을 대입한다.

$$\mathbf{x}_t + \delta\mathbf{x}_t = \mathbf{F}_t \delta\mathbf{x}_{t-1} + \mathbf{f}(\mathbf{x}_{t-1|t-1}, 0, \mathbf{u}_t, 0) + \mathbf{F}_w \mathbf{w}_t \quad (40)$$

nominal 상태 변수  $\mathbf{x}_t$ 는 정의 상 노이즈가 없는  $\mathbf{f}(\mathbf{x}_{t-1}, 0, \mathbf{u}_t, 0)$ 과 동일하므로 서로 소거된다.

$$\begin{aligned}\delta \mathbf{x}_t &= \mathbf{F}_t \delta \mathbf{x}_{t-1} + \mathbf{F}_w \mathbf{w}_t \\ &= \mathbf{F}_t \delta \mathbf{x}_{t-1} + \tilde{\mathbf{w}}_t \\ &= 0 + \tilde{\mathbf{w}}_t\end{aligned}\quad (41)$$

-  $\tilde{\mathbf{w}}_t = \mathbf{F}_w \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top)$

관측 모델 함수도 동일하게 에러 상태 변수의 값을 0으로 치환하면 아래와 같이 전개된다.

$$\begin{aligned}\mathbf{z}_t &= \mathbf{H}_t \delta \mathbf{x}_t + \mathbf{h}(\mathbf{x}_{t|t-1}, \delta \hat{\mathbf{x}}_{t-1|t-1}, 0) - \mathbf{H}_t \delta \hat{\mathbf{x}}_{t|t-1} + \mathbf{H}_v \mathbf{v}_t \\ &= \mathbf{h}(\mathbf{x}_{t|t-1}, 0, 0) + \mathbf{H}_v \mathbf{v}_t \\ &= \tilde{\mathbf{z}}_t + \tilde{\mathbf{v}}_t\end{aligned}\quad (42)$$

-  $\tilde{\mathbf{z}}_t = \mathbf{h}(\mathbf{x}_{t|t-1}, 0, 0) - \tilde{\mathbf{v}}_t = \mathbf{H}_v \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top)$

확률 변수가 모두 가우시안 분포를 따른다고 가정하면  $p(\delta \mathbf{x}_t | \delta \mathbf{x}_{t-1}, \mathbf{u}_t), p(\mathbf{z}_t | \delta \mathbf{x}_t)$ 은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}p(\delta \mathbf{x}_t | \delta \mathbf{x}_{t-1}, \mathbf{u}_t) &\sim \mathcal{N}(0, \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top) \\ &= \frac{1}{\sqrt{\det(2\pi \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top)}} \exp\left(-\frac{1}{2}(\delta \mathbf{x}_t)^\top (\mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top)^{-1} (\delta \mathbf{x}_t)\right)\end{aligned}\quad (43)$$

$$\begin{aligned}p(\mathbf{z}_t | \delta \mathbf{x}_t) &\sim \mathcal{N}(\tilde{\mathbf{z}}_t, \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top) \\ &= \frac{1}{\sqrt{\det(2\pi \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top)}} \exp\left(-\frac{1}{2}(\mathbf{z}_t - \tilde{\mathbf{z}}_t)^\top (\mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top)^{-1} (\mathbf{z}_t - \tilde{\mathbf{z}}_t)\right)\end{aligned}\quad (44)$$

$\mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top$ 은 선형화된 에러 상태 모션 모델의 노이즈를 의미하며  $\mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top$ 은 선형화된 관측 모델의 노이즈를 의미한다. 다음으로 칼만 필터를 통해 구해야 하는  $\overline{\text{bel}}(\delta \mathbf{x}_t), \text{bel}(\delta \mathbf{x}_t)$ 은 아래와 같이 나타낼 수 있다.

$$\begin{aligned}\overline{\text{bel}}(\delta \mathbf{x}_t) &= \int p(\delta \mathbf{x}_t | \delta \mathbf{x}_{t-1}, \mathbf{u}_t) \text{bel}(\delta \mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \sim \mathcal{N}(\delta \hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \\ \text{bel}(\delta \mathbf{x}_t) &= \eta \cdot p(\mathbf{z}_t | \delta \mathbf{x}_t) \overline{\text{bel}}(\delta \mathbf{x}_t) \sim \mathcal{N}(\delta \hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t})\end{aligned}\quad (45)$$

ESKF 또한 EKF와 동일하게 prediction에서 이전 스텝의 값과 모션 모델을 사용하여 예측값  $\overline{\text{bel}}(\delta \mathbf{x}_t)$ 을 먼저 구한 후 correction에서 관측값과 관측 모델을 사용하여 보정된 값  $\text{bel}(\delta \mathbf{x}_t)$ 을 구하는 방식으로 동작한다. 초기값  $\text{bel}(\delta \mathbf{x}_0)$ 은 다음과 같이 주어진다.

$$\text{bel}(\delta \mathbf{x}_0) \sim \mathcal{N}(0, \mathbf{P}_0) \quad (46)$$

-  $\delta \hat{\mathbf{x}}_0 = 0$  : 항상 0의 값을 가진다

-  $\mathbf{P}_0$  : 일반적으로 작은 값(<1e-2)으로 설정한다.

#### 4.1 Prediction step

Prediction은  $\overline{\text{bel}}(\delta \mathbf{x}_t)$ 를 구하는 과정을 말한다. 공분산 행렬을 구할 때 선형화된 자코비안 행렬  $\mathbf{F}_t$ 가 사용된다.

$$\begin{aligned}\delta \hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_t \delta \hat{\mathbf{x}}_{t-1|t-1} = 0 \quad \leftarrow \text{Always } 0 \\ \hat{\mathbf{x}}_{t|t-1} &= \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, 0, \mathbf{u}_t, 0) \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top\end{aligned}\quad (47)$$

위 식에서  $\mathbf{F}_t$ 는 에러 상태에 대한 자코비안임에 유의한다. 에러 상태 변수  $\delta \hat{\mathbf{x}}$ 는 매 correction 스텝이 끝나면 0으로 초기화된다. **여기서 선형 자코비안  $\mathbf{F}_t$ 를 곱해도 같은 0이 되기 때문에 prediction 스텝에서  $\delta \hat{\mathbf{x}}$  같은 항상 0이 된다.** 따라서 에러 상태  $\delta \hat{\mathbf{x}}_{t|t-1}$ 는 prediction 스텝에서는 변하지 않고 nominal 상태  $\hat{\mathbf{x}}_{t|t-1}$ 와 에러 상태의 공분산  $\mathbf{P}_{t|t-1}$ 만 업데이트된다.

## 4.2 Correction step

Correction은  $\text{bel}(\delta\mathbf{x}_t)$ 를 구하는 과정을 말한다. 칼만 게인과 공분산 행렬을 구할 때 선형화된 자코비안 행렬  $\mathbf{H}_t$ 가 사용된다.

$$\boxed{\begin{aligned}\mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v)^{-1} \\ \delta\hat{\mathbf{x}}_{t|t} &= \mathbf{K}_t (\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0, 0)) \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \delta\hat{\mathbf{x}}_{t|t} \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \\ \text{reset } \delta\hat{\mathbf{x}}_{t|t} &\\ \delta\hat{\mathbf{x}}_{t|t} &\leftarrow 0 \\ \mathbf{P}_{t|t} &\leftarrow \mathbf{G} \mathbf{P}_{t|t} \mathbf{G}^\top\end{aligned}} \quad (48)$$

위 식에서  $\mathbf{H}_t$ 는 관측 모델에 대한 true 상태  $\mathbf{x}_{\text{true},t}$ 의 자코비안이 아닌 에러 상태  $\delta\mathbf{x}_t$ 의 자코비안임에 유의한다.  $\mathbf{P}_{t|t-1}, \mathbf{K}_t$  또한 EKF와 기호만 같은 뿐 실제 값은 다름에 유의한다. 즉, 전체적인 공식은 EKF와 동일하지만  $\mathbf{F}, \mathbf{H}, \mathbf{P}, \mathbf{K}$  행렬이 에러 상태  $\delta\mathbf{x}_t$ 에 대한 값을 의미하는 점이 다르다.

### 4.2.1 Reset

nominal 상태가 정상적으로 업데이트되면 다음으로 에러 상태의 값을 0으로 리셋해야 한다. 리셋을 하는 이유는 새로운 nominal 상태에 대한 새로운 에러(new error)를 표현해야 하기 때문이다. 리셋으로 인해 에러 상태의 공분산  $\mathbf{P}_{t|t}$ 이 업데이트된다.

리셋 함수를  $\mathbf{g}(\cdot)$ 라고 하면 이는 다음과 같이 나타낼 수 있다. 이에 대한 자세한 내용은 Quaternion kinematics for the error-state Kalman filter 내용 정리 포스트의 챕터 6 내용을 참고하면 된다.

$$\delta\mathbf{x} \leftarrow \mathbf{g}(\delta\mathbf{x}) = \delta\mathbf{x} - \delta\hat{\mathbf{x}}_{t|t-1} \quad (49)$$

ESKF의 리셋 과정은 다음과 같다.

$$\boxed{\begin{aligned}\delta\hat{\mathbf{x}}_{t|t} &\leftarrow 0 \\ \mathbf{P}_{t|t} &\leftarrow \mathbf{G} \mathbf{P}_{t|t} \mathbf{G}^\top\end{aligned}} \quad (50)$$

$\mathbf{G}$ 는 다음과 같이 정의된 리셋에 대한 자코비안을 의미한다.

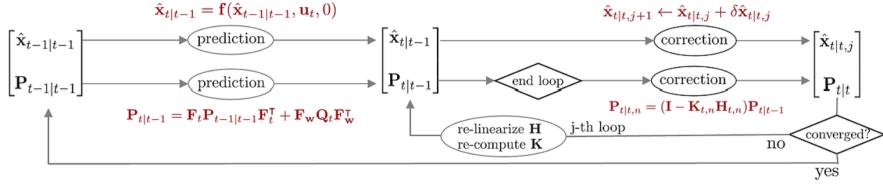
$$\mathbf{G} = \frac{\partial \mathbf{g}}{\partial \delta\mathbf{x}} \Big|_{\delta\mathbf{x}_t=\delta\hat{\mathbf{x}}_{t|t}} \quad (51)$$

## 4.3 Summary

ESKF를 함수로 표현하면 다음과 같다.

$$\boxed{\begin{aligned}&\text{ErrorStateKalmanFilter}(\hat{\mathbf{x}}_{t-1|t-1}, \delta\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1}, \mathbf{u}_t, \mathbf{z}_t)\{ \\ &\quad (\text{Prediction Step}) \\ &\quad \delta\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} = 0 \quad \leftarrow \text{Always 0} \\ &\quad \hat{\mathbf{x}}_{t|t-1} = \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, 0, \mathbf{u}_t, 0) \\ &\quad \mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top \\ \\ &\quad (\text{Correction Step}) \\ &\quad \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v)^{-1} \\ &\quad \delta\hat{\mathbf{x}}_{t|t} = \mathbf{K}_t (\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0, 0)) \\ &\quad \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \delta\hat{\mathbf{x}}_{t|t} \\ &\quad \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \\ &\quad \text{reset } \delta\hat{\mathbf{x}}_{t|t} \\ &\quad \delta\hat{\mathbf{x}}_{t|t} \leftarrow 0 \\ &\quad \mathbf{P}_{t|t} \leftarrow \mathbf{G} \mathbf{P}_{t|t} \mathbf{G}^\top \\ &\quad \text{return } \delta\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}\end{aligned}} \quad (52)$$

## 5 Iterated extended kalman filter (IEKF)



Iterated extended kalman filter (IEKF)는 EKF에서 correction 스텝 부분을 반복적으로 수행하는 알고리즘이다. EKF는 비선형 함수를 선형화하여 상태 변수를 추정하기 때문에 선형화 과정에서 필연적으로 오차가 발생할 수밖에 없다. IEKF는 이러한 선형화 오차를 줄이기 위해서 correction 스텝이 종료된 후 업데이트 변화량  $\delta \hat{x}_{t|t,j}$ 이 충분히 크다면 다시 선형화를 수행하여 반복적으로 correction 스텝을 진행한다.

이 때,  $\delta \hat{x}$ 는 문맥 상 예상 상태 변수가 아닌 업데이트 변화량으로 해석된다. 즉  $\hat{x}_{t|t,j+1} \leftarrow \hat{x}_{t|t,j} + \delta \hat{x}_{t|t,j}$ 와 같이 j번째 posterior 값에 더해져서 j+1 번째 posterior 값을 업데이트하는 용도로만 사용된다. 다시 말하면, 상태 추정의 대상이 아니다.

### 5.1 Compare to EKF

#### 5.1.1 Commonality 1

IEKF에서 모션 모델과 관측 모델은 다음과 같다. 이는 EKF와 완전히 동일하다.

$$\begin{aligned} \text{Motion Model: } & \mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{w}_t) \\ \text{Observation Model: } & \mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t) \end{aligned} \quad (53)$$

-  $\mathbf{x}_t$ : 모델의 상태 변수(state variable) -  $\mathbf{u}_t$ : 모델의 입력(input) -  $\mathbf{z}_t$ : 모델의 관측값(measurement) -  $\mathbf{f}(\cdot)$ : 비선형 모션(motion) 모델 함수 -  $\mathbf{h}(\cdot)$ : 비선형 관측(observation) 모델 함수 -  $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ : 모션 모델의 노이즈.  $\mathbf{Q}_t$ 는  $\mathbf{w}_t$ 의 공분산 행렬을 의미 -  $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ : 관측 모델의 노이즈.  $\mathbf{R}_t$ 는  $\mathbf{v}_t$ 의 공분산 행렬을 의미

#### 5.1.2 Commonality 2

다음으로 선형화 과정도 EKF의 (24), (25)와 완전히 동일하다.

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) - \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{F}_w \mathbf{w}_t \\ &= \mathbf{F}_t \mathbf{x}_{t-1} + \tilde{\mathbf{u}}_t + \tilde{\mathbf{w}}_t \end{aligned} \quad (54)$$

$$- \tilde{\mathbf{u}}_t = \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) - \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} - \tilde{\mathbf{w}}_t = \mathbf{F}_w \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^T)$$

$$\begin{aligned} \mathbf{z}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} + \mathbf{H}_v \mathbf{v}_t \\ &= \mathbf{H}_t \mathbf{x}_t + \tilde{\mathbf{z}}_t + \tilde{\mathbf{v}}_t \end{aligned} \quad (55)$$

$$- \tilde{\mathbf{z}}_t = \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} - \tilde{\mathbf{v}}_t = \mathbf{H}_v \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^T)$$

#### 5.1.3 Commonality 3

IEKF는 EKF와 동일하게  $\mathbf{F}, \mathbf{H}, \mathbf{K}$  행렬이 true 상태 변수  $\mathbf{x}_{\text{true}}$ 에 대한 행렬을 사용한다.

#### 5.1.4 Difference 1

**EKF**: prediction 값으로부터 한 번에 correction 값이 나온다.

**IEKF**: correction 값이 다시 prediction 값이 되어 correction 스텝을 반복적으로 진행한다.

$$\begin{aligned} \text{EKF Correction : } & \hat{\mathbf{x}}_{t|t-1} \rightarrow \hat{\mathbf{x}}_{t|t} \\ \text{IEKF Correction : } & \hat{\mathbf{x}}_{t|t,j} \leftarrow \hat{\mathbf{x}}_{t|t,j+1} \end{aligned} \quad (56)$$

-  $j$ : j-th iteration

### 5.1.5 Difference 2

(55) 식을 전개하여 innovation term  $\mathbf{r}_t$ 을 만들어보면 다음과 같다.

$$\mathbf{r}_t = \mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) - \mathbf{H}_t(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) \quad (57)$$

EKF의 경우 correction 스텝 (31)의 두 번째 줄을 보면 칼만 게인  $\mathbf{K}_t$ 에  $\mathbf{r}_t$ 가 곱해져서 posterior를 계산하는 것을 볼 수 있다.

$$\begin{aligned} \text{EKF correction : } \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{r}_t) \\ &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0)) \end{aligned} \quad (58)$$

**EKF**:  $\mathbf{H}_t(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1})$  부분은  $\mathbf{x}_t = \hat{\mathbf{x}}_{t|t-1}$ 이 대입되어 소거되고 나머지 부분만 사용된다.

**IEKF**: 매 순간 correction 스텝을 반복하면서 새로운 상태(new operating point)에 대한 선형화를 수행하기 때문에 해당 부분이 소거되지 않는다.

$$\begin{aligned} \text{EKF innovation term : } \mathbf{r}_t &= \mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, 0) \\ \text{IEKF innovation term : } \mathbf{r}_{t,j} &= \mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t,j}, 0) - \mathbf{H}_{t,j}(\hat{\mathbf{x}}_{t|t-1} - \hat{\mathbf{x}}_{t|t,j}) \end{aligned} \quad (59)$$

만약 첫 번째 iteration  $j = 0$ 의 경우  $\hat{\mathbf{x}}_{t|t,0} = \hat{\mathbf{x}}_{t|t-1}$ 이므로 소거되어 EKF와 동일한 식이 되지만  $j = 1$  이상부터는 다른 값이 되므로 소거되지 않는다.

## 5.2 Prediction step

Prediction은  $\overline{\text{bel}}(\mathbf{x}_t)$ 를 구하는 과정을 말한다. 공분산 행렬을 구할 때 선형화된 자코비안 행렬  $\mathbf{F}_t$ 가 사용된다. 이는 EKF의 prediction 스텝과 완전히 동일하다.

$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top \end{aligned} \quad (60)$$

## 5.3 Correction step

Correction은  $\text{bel}(\mathbf{x}_t)$ 를 구하는 과정을 말한다. 칼만 게인과 공분산 행렬을 구할 때 선형화된 자코비안 행렬  $\mathbf{H}_t$ 가 사용된다. IEKF의 correction 스텝은 업데이트 변화량  $\delta\hat{\mathbf{x}}_{t|t,j}$ 가 충분히 작아질 때까지 반복적(iterative)으로 수행한다.

```

set ε
start j-th loop
     $\mathbf{K}_{t,j} = \mathbf{P}_{t|t-1} \mathbf{H}_{t,j}^\top (\mathbf{H}_{t,j} \mathbf{P}_{t|t-1} \mathbf{H}_{t,j}^\top + \mathbf{H}_{v,j} \mathbf{R}_t \mathbf{H}_{v,j}^\top)^{-1}$ 
     $\delta\hat{\mathbf{x}}_{t|t,j} = \mathbf{K}_{t,j} (\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t,j}, 0) - \mathbf{H}_t(\hat{\mathbf{x}}_{t|t-1} - \hat{\mathbf{x}}_{t|t,j}))$ 
     $\hat{\mathbf{x}}_{t|t,j+1} = \hat{\mathbf{x}}_{t|t,j} + \delta\hat{\mathbf{x}}_{t|t,j}$ 
    iterate until  $\delta\hat{\mathbf{x}}_{t|t,j} < \epsilon$ .
end loop
 $\mathbf{P}_{t|t,n} = (\mathbf{I} - \mathbf{K}_{t,n} \mathbf{H}_{t,n}) \mathbf{P}_{t|t-1}$ 

```

## 5.4 Summary

IEKF를 함수로 표현하면 다음과 같다.

---

```

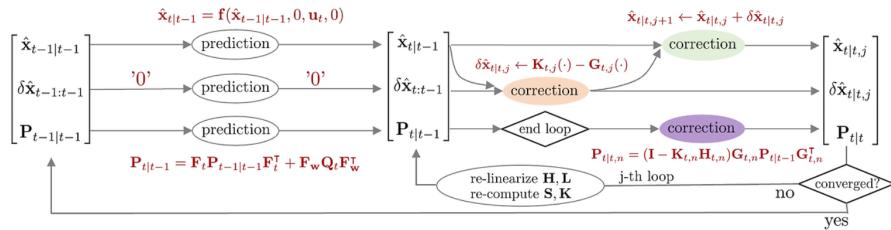
IteratedExtendedKalmanFilter( $\hat{\mathbf{x}}_{t-1|t-1}$ ,  $\mathbf{P}_{t-1|t-1}$ ,  $\mathbf{u}_t$ ,  $\mathbf{z}_t$ ){
    (Prediction Step)
     $\hat{\mathbf{x}}_{t|t-1} = \mathbf{f}_t(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0)$ 
     $\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top$ 

    (Correction Step)
    set  $\epsilon$ 
    start j-th loop
         $\mathbf{K}_{t,j} = \mathbf{P}_{t|t-1} \mathbf{H}_{t,j}^\top (\mathbf{H}_{t,j} \mathbf{P}_{t|t-1} \mathbf{H}_{t,j}^\top + \mathbf{H}_{v,j} \mathbf{R}_t \mathbf{H}_{v,j}^\top)^{-1}$ 
         $\delta\hat{\mathbf{x}}_{t|t,j} = \mathbf{K}_{t,j} (\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t,j}, 0) - \mathbf{H}_t (\hat{\mathbf{x}}_{t|t-1} - \hat{\mathbf{x}}_{t|t,j}))$ 
         $\hat{\mathbf{x}}_{t|t,j+1} = \hat{\mathbf{x}}_{t|t,j} + \delta\hat{\mathbf{x}}_{t|t,j}$ 
        iterate until  $\delta\hat{\mathbf{x}}_{t|t,j} < \epsilon$ .
    end loop
     $\mathbf{P}_{t|t,n} = (\mathbf{I} - \mathbf{K}_{t,n} \mathbf{H}_{t,n}) \mathbf{P}_{t|t-1}$ 
    return  $\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}$ 
}

```

(62)

## 6 Iterated error-state kalman filter (IESKF)



## 7 Relation between Gauss-Newton and IEKF

## 8 Wrap-up

지금까지 설명한 KF, EKF, ESKF, IEKF, IESKF를 한 장의 슬라이드로 표현하면 다음과 같다. 클릭하면 큰 그림으로 볼 수 있다.

## 8.1 Kalman Filter (KF)

### Kalman Filter(KF)

Motion Model:  $\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t$   
 Observation Model:  $\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t$

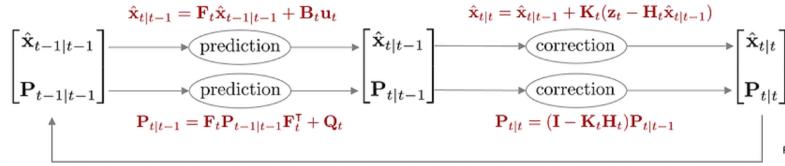
$$\begin{aligned}\overline{\text{bel}}(\mathbf{x}_t) &\sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \\ \text{bel}(\mathbf{x}_t) &\sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \\ \mathbf{w}_t &\sim \mathcal{N}(0, \mathbf{Q}_t) \\ \mathbf{v}_t &\sim \mathcal{N}(0, \mathbf{R}_t)\end{aligned}$$

EKF Prediction:

$$\begin{aligned}\hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t\end{aligned}$$

EKF Correction:

$$\begin{aligned}\mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_t)^{-1} \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}) \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}\end{aligned}$$



For more information, please visit [alida.tistory.com](http://alida.tistory.com)

## 8.2 Extended Kalman Filter (EKF)

### Extended Kalman Filter(EKF)

Motion Model:  $\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{w}_t) \simeq \mathbf{F}_t \mathbf{x}_{t-1} + \tilde{\mathbf{u}}_t + \tilde{\mathbf{w}}_t$   
 Observation Model:  $\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t) \simeq \mathbf{H}_t \mathbf{x}_t + \tilde{\mathbf{z}}_t + \tilde{\mathbf{v}}_t$

$$\begin{aligned}\overline{\text{bel}}(\mathbf{x}_t) &\sim \mathcal{N}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1}) \\ \text{bel}(\mathbf{x}_t) &\sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \\ \tilde{\mathbf{w}}_t &\sim \mathcal{N}(0, \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top) \\ \tilde{\mathbf{v}}_t &\sim \mathcal{N}(0, \mathbf{H}_v \mathbf{R}_t \mathbf{H}_v^\top)\end{aligned}$$

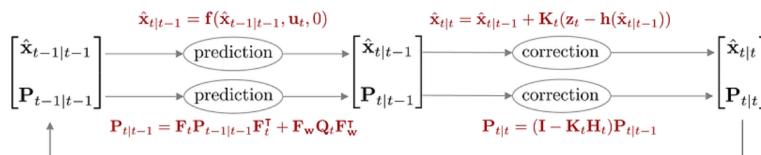
EKF Prediction:

$$\begin{aligned}\hat{\mathbf{x}}_{t|t-1} &= \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}, \mathbf{u}_t, 0) \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{F}_w \mathbf{Q}_t \mathbf{F}_w^\top\end{aligned}$$

$$\begin{aligned}\mathbf{F}_t &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_{t-1}} \right|_{\mathbf{x}_{t-1}=\hat{\mathbf{x}}_{t-1|t-1}} \\ \mathbf{F}_w &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}_t} \right|_{\substack{\mathbf{x}_{t-1}=\hat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{w}_t=0}} \\ \mathbf{H}_t &= \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_t} \right|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}} \\ \mathbf{H}_v &= \left. \frac{\partial \mathbf{h}}{\partial \mathbf{v}_t} \right|_{\substack{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1} \\ \mathbf{v}_t=0}}\end{aligned}$$

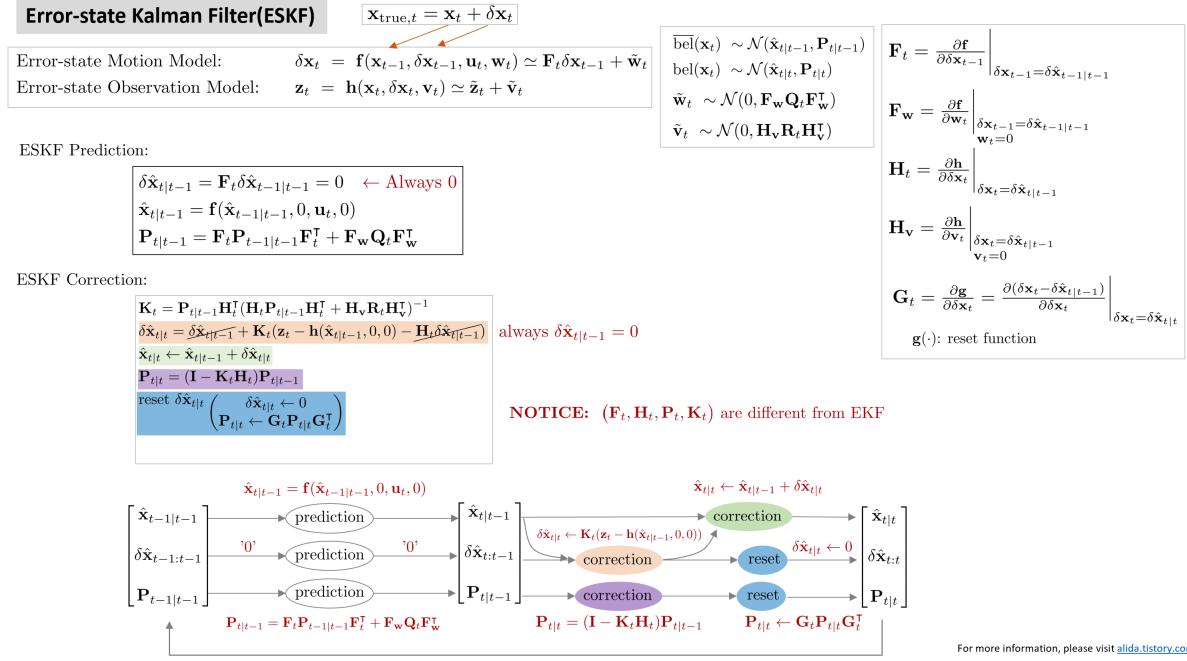
EKF Correction:

$$\begin{aligned}\mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_v \mathbf{R}_v^\top)^{-1} \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1})) \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}\end{aligned}$$

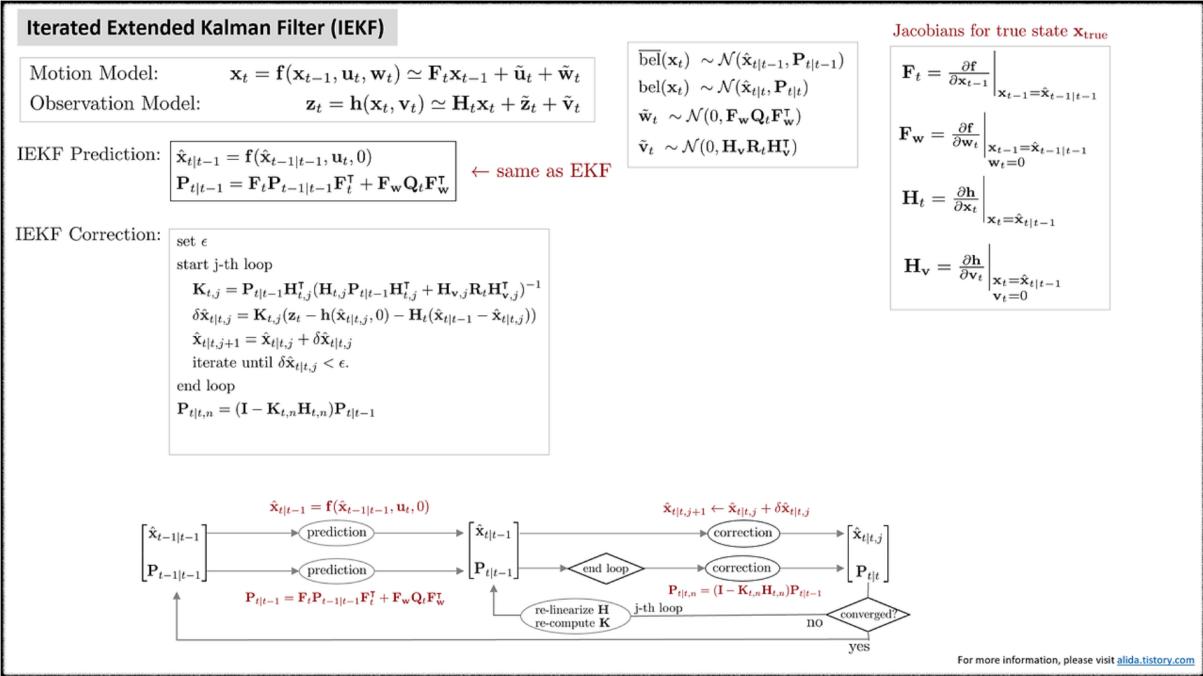


For more information, please visit [alida.tistory.com](http://alida.tistory.com)

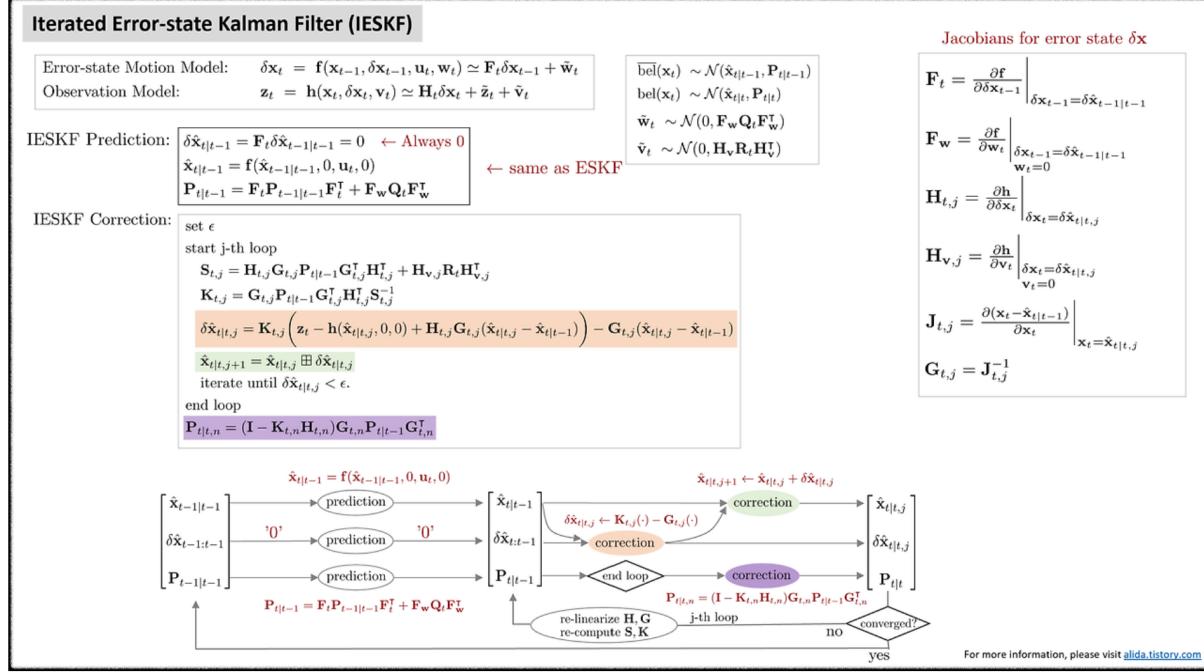
### 8.3 Error-state Kalman Filter (ESKF)



### 8.4 Iterated Extended Kalman Filter (IEKF)



## 8.5 Iterated Error-state Kalman Filter (IESKF)



## 9 Reference

- [1] (Wiki) Kalman Filter
- [2] (Paper) Sola, Joan. "Quaternion kinematics for the error-state Kalman filter." arXiv preprint arXiv:1711.02508 (2017).
- [3] (Youtube) Robot Mapping Coure - Freiburg Univ
- [4] (Blog) [SLAM] Kalman filter and EKF(Extended Kalman Filter) - Jinyong Jeong
- [5] (Blog) Error-State Kalman Filter understanding and formula derivation - CSDN
- [6] (Paper) He, Dongjiao, Wei Xu, and Fu Zhang. "Kalman filters on differentiable manifolds." arXiv preprint arXiv:2102.03804 (2021).
- [7] (Book) SLAM in Autonomous Driving book (SAD book)
- [8] (Paper) Xu, Wei, and Fu Zhang. "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter." IEEE Robotics and Automation Letters 6.2 (2021): 3317-3324.
- [9] (Paper) Huai, Jianzhu, and Xiang Gao. "A Quick Guide for the Iterated Extended Kalman Filter on Manifolds." arXiv preprint arXiv:2307.09237 (2023).
- [10] (Paper) Bloesch, Michael, et al. "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback." The International Journal of Robotics Research 36.10 (2017): 1053-1072.

## 10 Revision log

- 1st: 2020-06-23
- 3rd: 2020-06-26
- 5th: 2023-01-31
- 7th: 2023-02-04
- 2nd: 2020-06-24
- 4th: 2023-01-21
- 6th: 2023-02-02
- 8th: 2024-02-08